

Using virtualenv to Manage Your Own Python Environment

NAS provides several Python environments including the system default, `/usr/bin/python3` (currently version 3.6.8), and modules such as `python3/3.9.5`, and so on. These environments provide a variety of Python packages and versions. However, if you need a Python package or version that is not available in these environments, we recommend that you install it in your own directory.

To install a Python package or version, we recommend using the `virtualenv` tool, which provides these benefits:

- It creates an isolated Python virtual environment. When activated, you can install and use packages through this environment, instead of through `/usr/bin/python` or the NAS-provided Python modules.
- You can create separate virtual environments for separate projects that may need different versions of the same package.

Complete these steps to check whether a package is available on NAS systems, create a virtual environment, install a package from the Python Package Index (PyPI), and use the package in this virtual environment:

1. Load the latest Python module (currently `python3/3.9.5`), which already has `virtualenv`, `pip` and many other Python packages installed:

```
% module load python3/3.9.5
```

2. View the packages and versions that are available under `python3/3.9.5`:

```
% pip list
```

In the output, look for a package with the version you want. If it is not available, or if you want to install your own private copy, continue with the steps below.

3. Create a virtual environment with a new directory name, such as `~/myenv`. The `virtualenv` tool will create the directory for you.

If you want this virtual environment to be able to access packages already available in the global site-packages (in this example, `/nasa/pkgsrc/toss3/2021Q2/lib/python3.9/site-packages`), so that you do not have to install everything from scratch, run:

```
% virtualenv --system-site-packages ~/myenv
```

Otherwise, run:

```
% virtualenv ~/myenv
```

At the end of this step, you should see the `python` (which is linked to `python3.9`), `pip` (which is same as `pip3` or `pip3.9`), `wheel`, and multiple activate scripts under `~/myenv/bin`.

4. Activate the virtual environment:

```
% source ~/myenv/bin/activate (for bash)
[myenv]%
```

or

```
% source ~/myenv/bin/activate.csh (for csh)
[myenv]%
```

Your prompt should now include the virtual environment `[myenv]`.

5. Use the **pip** tool from this environment to install or upgrade a package (for example, SymPy) under **~/myenv/lib/python3.9/site-packages**, and run the **pip list** command to check whether the installation is completed properly. (Note that SymPy depends on **mpmath**, so make sure that **mpmath** is installed before installing **sympy**.)

```
[myenv]% which pip
/u/username/myenv/bin/pip
[myenv]% pip install --upgrade sympy
[myenv]% pip list
```

6. When you finish using this isolated environment, run the deactivate command to exit. After you exit, your prompt should no longer include **[myenv]**.

```
[myenv]% deactivate
%
```

To install another package or to use a package in this isolated virtual environment at a later time, you do not need to load the **python/3.9.5** module again. Simply activate the virtual environment (step 4, above), then use the packages or install additional packages. For example:

```
% source ~/myenv/bin/activate (for bash)

or

% source ~/myenv/bin/activate.csh (for csh)

[myenv]% pip install --upgrade pendulum
[myenv]% pip list
[myenv]% which python
/u/username/myenv/bin/python
[myenv]% python
>>> import sympy as s
>>> import pendulum as p
```

Article ID: 540

Last updated: 18 Feb, 2022

Revision: 59

Filesystems & Software -> Software -> Using virtualenv to Manage Your Own Python Environment

<https://www.nas.nasa.gov/hecc/support/kb/entry/540/>